

# **Toteutusdokumentaatio**

## **Shakki-AI**

### **Sebastian Falk**

Kaikki ohjelman koodit ovat paketissa AI, kaikki muut paketit sisältävät asioita jotka eivät kuulu tähän projektiin vaan itse shakkipeliin. AI paketti sisältää kolme luokkaa: DeepShoe, Evaluator ja MoveCreator. DeepShoe tekee kaiken varsinaisen työn, se etsii parhaan siirron kun bestMove metodia kutsutaan käyttämällä alphaBeta karsintaa. Evaluator luokka suorittaa pelilaudan arvioinnin evaluointifunktion avulla, se kertoo mikä pelilauta on paras antamalla pelilaudalle arvon. MoveCreator luokkaa käytetään DeepShoe luokassa. Käyttämällä sitä saadaan lista kaikista siirroista tietyille värille, joko järjestettynä tai ei.

Ohjelman aikavaativuus on  $O(M^N)$  ja tilavaativuus on sama. Missä M on mahdollisten siirtojen määrä ja N on haluttu syvyys alphaBeta haussa. Tästä syystä on aika selvää että suuri syvyys vaatii paljon aikaa. Olen itse huomannut että syvyydellä 4 kestää hetki saada siirto mutta ei kuitenkaan liian kauan. Jos syvyys on 5 tai suurempi niin kestää liian kauan aikaa että sitä jaksaisi odottaa. Eli syvyys 3 on aika sopiva, silloin ei kestä ollenkaan kauan eikä siirto ole liian ”huono”.

Olen aika tyytyväinen ohjelmaani mutta se ei kuitenkaan ole maailman parhain. Se ei tullut minulle suurena yllätyksenä että tekoälyn tekeminen on aika vaikeaa. Tekoäly voittaa varmaan vähän huonommat pelaajat mutta ei varmaan hyviä pelaajia. Löytyy paljon tapoja joilla ohjelmaa pystyisi parantamaan mutta ne olivat hieman liian vaikeita minulle. Jos minulla olisi enemmän aikaa ja taitoja niin olisin voinut käyttää transposition tablesia, principal variation searchiä tai jopa koneoppimista. Tämä kerta en käyttänyt näitä mutta ehkä voisin parantaa ohjelmaani joskus tulevaisuudessa.

Lähteitä:

- <http://chessprogramming.wikispaces.com/Alpha-Beta>
- <https://chessprogramming.wikispaces.com/Evaluation>